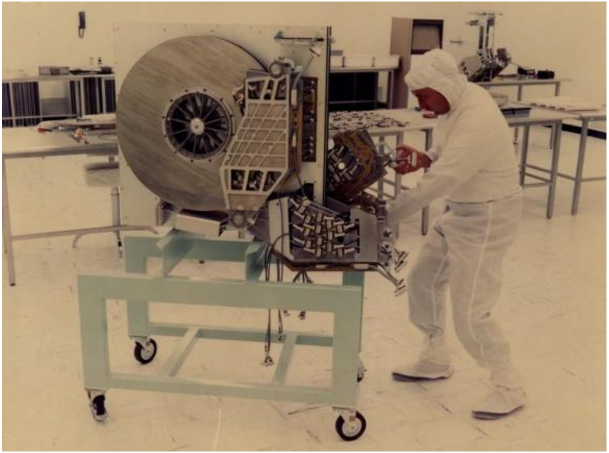
# 3-d stabilizer codes with a power law energy barrier:1208.3496

Kamil Michnicki

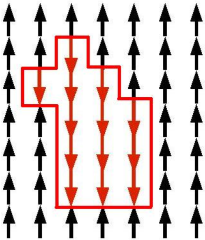
University of Washington, Department of Physics

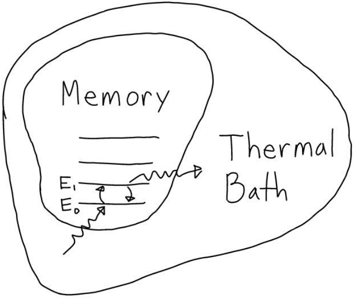
January 23, 2013

# Classical self-correcting memories



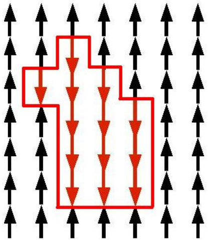
# Self-correcting memory

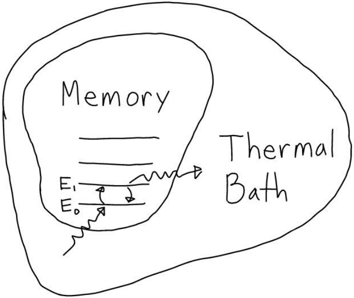





- Local interactions provide a force towards a stable phase.

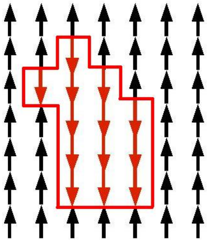
# Self-correcting memory

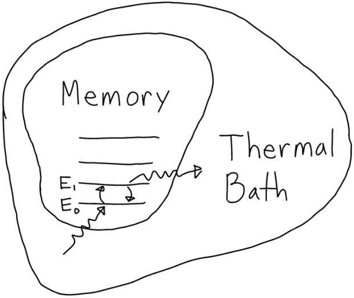





- Local interactions provide a force towards a stable phase.
- Low-temperature bath dissipates heat.

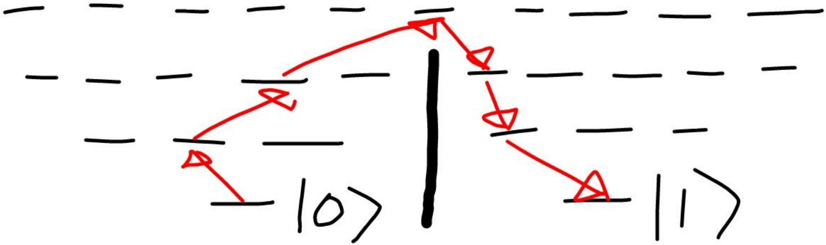
# Self-correcting memory





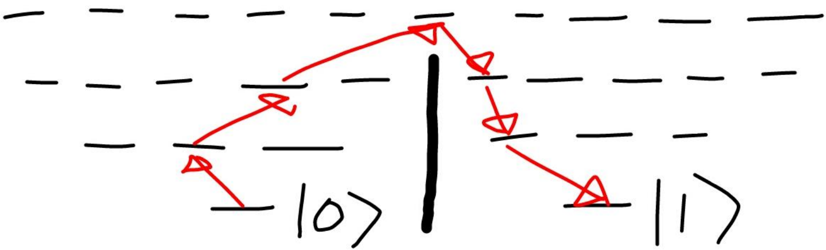

- Local interactions provide a force towards a stable phase.
- Low-temperature bath dissipates heat.
- This provides a bias towards the lower energies, typically modeled as $r(E_a \to E_b) \propto e^{-\beta(E_b - E_a)}$.

# Two conditions for self-correcting memory



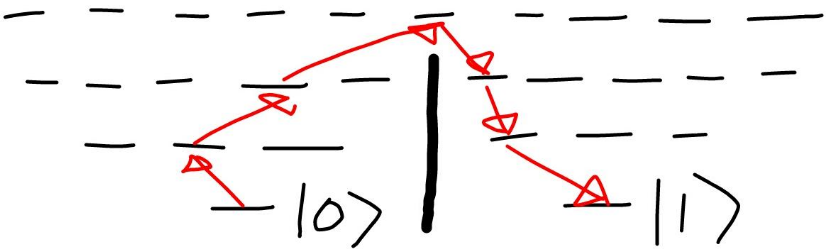

- We want the energy barrier per path to be high

# Two conditions for self-correcting memory



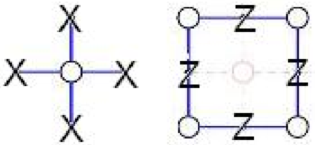

- We want the energy barrier per path to be high
- We want the number of paths leading to a logical error to be low

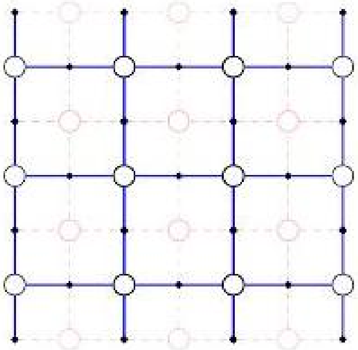
# Two conditions for self-correcting memory




- We want the energy barrier per path to be high
- We want the number of paths leading to a logical error to be low
- *Main Result:* An exponential improvement for the energy barrier of local stabilizer code Hamiltonians in 3-d compared to the previous best.

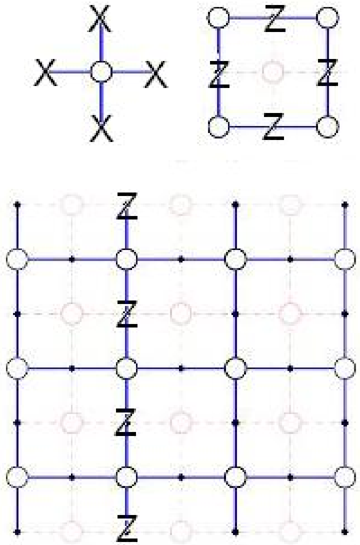
# Toric code/surface code [Kitaev '97]



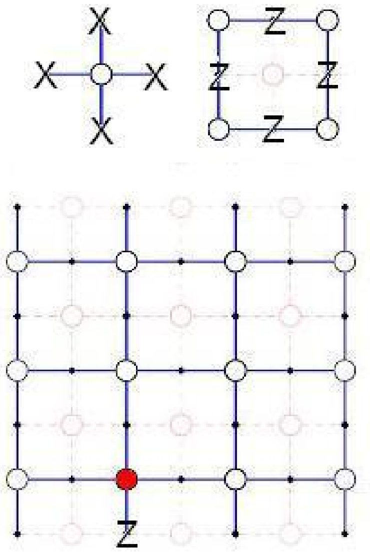
- Error correction at the physical level
- $H = -\sum_{h \in R} h$, where $R$ is a local generating set for the stabilizer group

# Toric code [Kitaev '97]



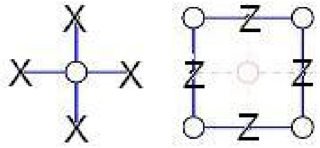

- Error correction at the physical level
- $H = -\sum_{h \in R} h$, where $R$ is a local generating set for the stabilizer group
- Large distance $O(\sqrt{n})$.

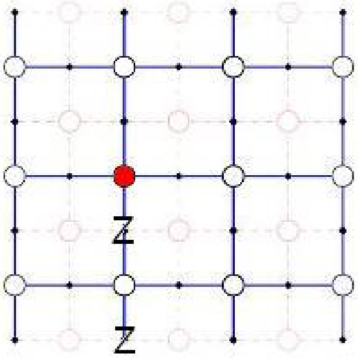
# Toric code [Kitaev '97]




- Error correction at the physical level
- $H = -\sum_{h \in R} h$, where $R$ is a local generating set for the stabilizer group
- Large distance $O(\sqrt{n})$.
- Constant energy barrier.

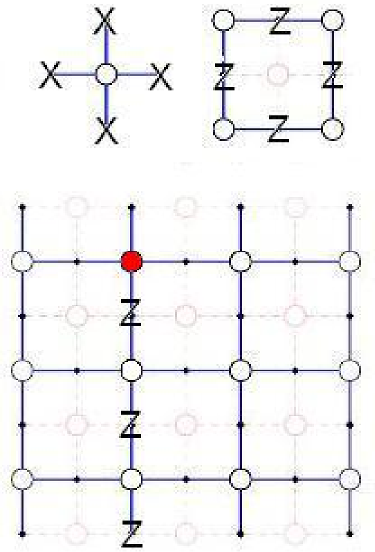
# Toric code [Kitaev '97]



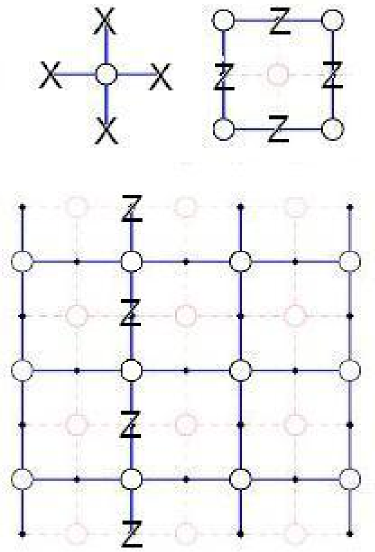

- Error correction at the physical level
- $H = -\sum_{h \in R} h$, where $R$ is a local generating set for the stabilizer group
- Large distance $O(\sqrt{n})$.
- Constant energy barrier.

# Toric code [Kitaev '97]




- Error correction at the physical level
- $H = -\sum_{h \in R} h$, where $R$ is a local generating set for the stabilizer group
- Large distance $O(\sqrt{n})$.
- Constant energy barrier.

# Toric code [Kitaev '97]




- Error correction at the physical level
- $H = -\sum_{h \in R} h$, where $R$ is a local generating set for the stabilizer group
- Large distance $O(\sqrt{n})$.
- Constant energy barrier.

# Previous work

- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]

# Previous work

- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]
- No-go theorem in 3-d for translation and scale-invariant stabilizer Hamiltonians [Yoshida '11] .

# Previous work

- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]
- No-go theorem in 3-d for translation and scale-invariant stabilizer Hamiltonians [Yoshida '11] .
- 4-d toric code is known to be self correcting [Alicki et al. '10].

# Previous work

- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]
- No-go theorem in 3-d for translation and scale-invariant stabilizer Hamiltonians [Yoshida '11] .
- 4-d toric code is known to be self correcting [Alicki et al. '10].
- Haah code breaks scale invariance. $\delta E \sim \log(L)$ [Haah '11]

# Previous work

- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]
- No-go theorem in 3-d for translation and scale-invariant stabilizer Hamiltonians [Yoshida '11] .
- 4-d toric code is known to be self correcting [Alicki et al. '10].
- Haah code breaks scale invariance. $\delta E \sim \log(L)$ [Haah '11]
- **Welded solid code breaks translation invariance.** $\delta E \sim L^{\frac{2}{3}}$**.**

# Previous work

- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]
- No-go theorem in 3-d for translation and scale-invariant stabilizer Hamiltonians [Yoshida '11] .
- 4-d toric code is known to be self correcting [Alicki et al. '10].
- Haah code breaks scale invariance. $\delta E \sim \log(L)$ [Haah '11]
- **Welded solid code breaks translation invariance.** $\delta E \sim L^{\frac{2}{3}}$**.**
- Lower bound on the storage time $t \sim \frac{e^{\beta \delta E}}{N} 2^{-k(L)}$ for a maximum system size of $N \lesssim e^{\beta}$ [Bravyi-Haah '11].

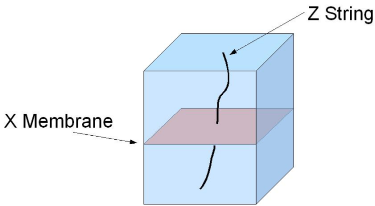
# Previous work

- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]
- No-go theorem in 3-d for translation and scale-invariant stabilizer Hamiltonians [Yoshida '11] .
- 4-d toric code is known to be self correcting [Alicki et al. '10].
- Haah code breaks scale invariance. $\delta E \sim \log(L)$ [Haah '11]
- **Welded solid code breaks translation invariance. $\delta E \sim L^{\frac{2}{3}}$.**
- Lower bound on the storage time $t \sim \frac{e^{\beta \delta E}}{N} 2^{-k(L)}$ for a maximum system size of $N \lesssim e^{\beta}$ [Bravyi-Haah '11].
    - The Haah code gives $t \gtrsim \exp(c\beta^2)$

# Previous work

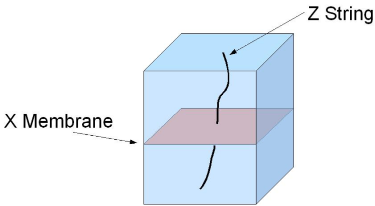
- No-go theorems in 2-d for stabilizer Hamiltonians. [Bravyi-Terhal '08]
- No-go theorem in 3-d for translation and scale-invariant stabilizer Hamiltonians [Yoshida '11] .
- 4-d toric code is known to be self correcting [Alicki et al. '10].
- Haah code breaks scale invariance. $\delta E \sim \log(L)$ [Haah '11]
- **Welded solid code breaks translation invariance.** $\delta E \sim L^{\frac{2}{3}}$**.**
- Lower bound on the storage time $t \sim \frac{e^{\beta \delta E}}{N} 2^{-k(L)}$ for a maximum system size of $N \lesssim e^{\beta}$ [Bravyi-Haah '11].
    - The Haah code gives $t \gtrsim \exp(c\beta^2)$
    - The welded solid code gives $t \gtrsim \exp(\exp(\frac{2}{9}\beta))$

# Welded solid codes



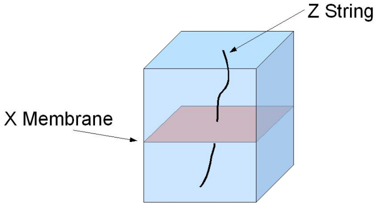

- Solid code=3-d toric code with smooth & rough boundaries.

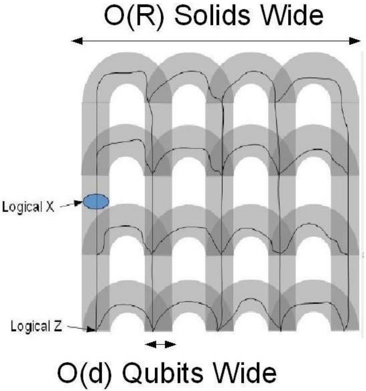
# Welded solid codes




- Solid code=3-d toric code with smooth & rough boundaries.
- $\bar{Z}$ is a string operator with a constant energy barrier.
- $\bar{X}$ is a membrane operator with an $O(d)$ energy barrier.

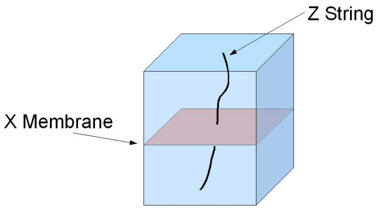
# Welded solid codes

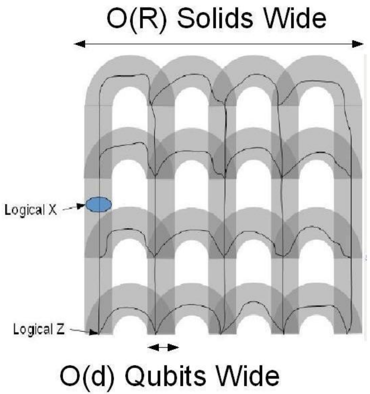



- We can increase the energy barrier of the strings by welding them together.

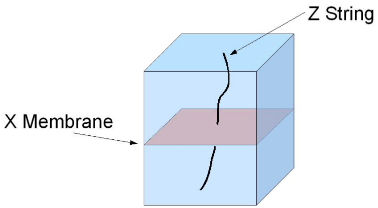
# Welded solid codes

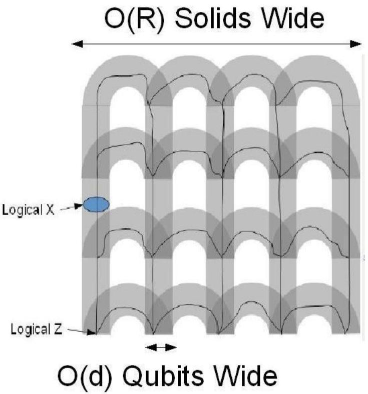



- $O(R)$ solids wide & $O(d)$ qubits wide per solid $\rightarrow N \sim R^2 d^3$

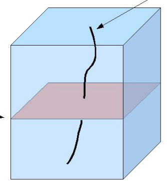
# Welded solid codes

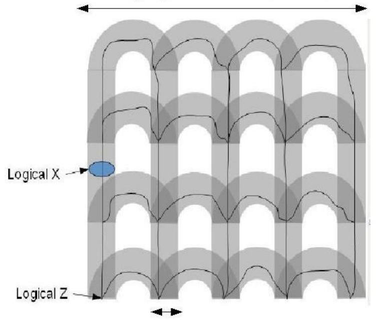



- $O(R)$ solids wide & $O(d)$ qubits wide per solid $\rightarrow N \sim R^2 d^3$
- The bifurcating string operator has an $O(R)$ energy barrier.

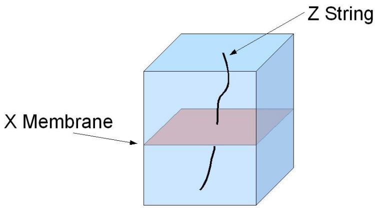
# Welded solid codes

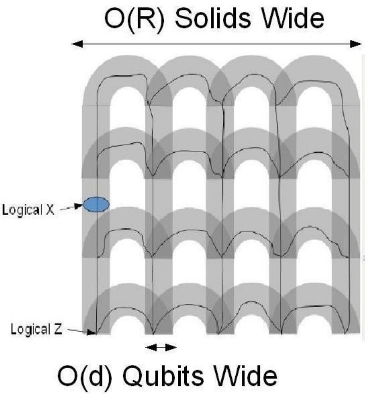



- $O(R)$ solids wide & $O(d)$ qubits wide per solid $\rightarrow N \sim R^2 d^3$
- The bifurcating string operator has an $O(R)$ energy barrier.
- The membrane operator still has an $O(d)$ energy barrier.

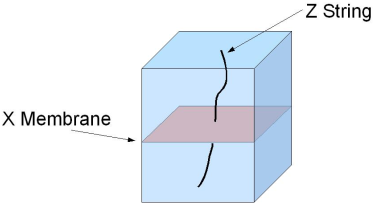
# Welded solid codes

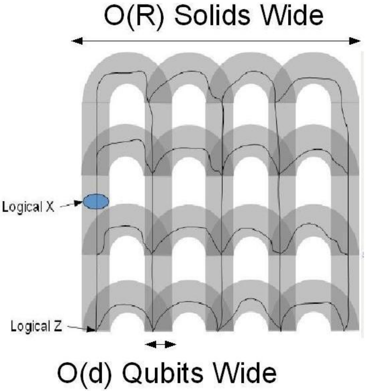



- $O(R)$ solids wide & $O(d)$ qubits wide per solid $\rightarrow N \sim R^2 d^3$
- The bifurcating string operator has an $O(R)$ energy barrier.
- The membrane operator still has an $O(d)$ energy barrier.
- $\delta E = min(d, R)$, which maximizes the energy barrier per qubit when $d = R \rightarrow N \sim d^5 \sim \delta E^5 \rightarrow \delta E \sim N^{\frac{1}{5}} \sim L^{\frac{3}{5}}$.

# Welded solid codes




- Using a 3-d lattice for the outer blocks, $\delta E = N^{\frac{2}{9}} = L^{\frac{2}{3}}$

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{matrix} I & X & X \\ X & I & X \end{matrix}$$

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{array}{ccc} I & X & X \\ X & I & X \\ Z & Z & Z \end{array}$$

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$
\begin{array}{cccccc}
I & X & X & & & \\
X & I & X & & & \\
Z & Z & Z & & & \\
 & & X & I & I & X \\
 & & X & X & X & X
\end{array}
$$

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{matrix} I & X & X & & & \\ X & I & X & & & \\ Z & Z & Z & & & \\ & & X & I & I & X \\ & & X & X & X & X \\ & & Z & I & I & Z \end{matrix}$$

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$
\begin{array}{cccccc}
I & X & X & & & \\
X & I & X & & & \\
Z & Z & Z & & & \\
 & & X & I & I & X \\
 & & X & X & X & X \\
 & & Z & I & I & Z
\end{array}
$$

$$
\begin{array}{cccccc}
I & X & X & I & I & I \\
X & I & X & I & I & I \\
I & I & X & I & I & X \\
I & I & X & X & X & X
\end{array}
$$

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{matrix} I & X & X & & & \\ X & I & X & & & \\ Z & Z & Z & & & \\ & & X & I & I & X \\ & & X & X & X & X \\ & & Z & I & I & Z \end{matrix}$$

$$\begin{matrix} I & X & X & I & I & I \\ X & I & X & I & I & I \\ I & I & X & I & I & X \\ I & I & X & X & X & X \\ Z & Z & Z & I & I & Z \end{matrix}$$

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{matrix} I & X & X \\ X & I & X \\ Z & Z & Z \end{matrix}$$

$$\begin{matrix} X & I & I & X \\ X & X & X & X \\ Z & I & I & Z \end{matrix}$$

$$\begin{matrix} I & X & X & I & I & I \\ X & I & X & I & I & I \\ I & I & X & I & I & X \\ I & I & X & X & X & X \\ Z & Z & Z & I & I & Z \end{matrix}$$

1. Identify qubits between two codes.

# Welding

Example of welding two stabilizer codes with a Z-weld:

| | | | | | |
|---|---|---|---|---|---|
| $I$ | $X$ | $X$ | | | |
| $X$ | $I$ | $X$ | | | |
| $Z$ | $Z$ | $Z$ | | | |
| | | $X$ | $I$ | $I$ | $X$ |
| | | $X$ | $X$ | $X$ | $X$ |
| | | $Z$ | $I$ | $I$ | $Z$ |

| | | | | | |
|---|---|---|---|---|---|
| $I$ | $X$ | $X$ | $I$ | $I$ | $I$ |
| $X$ | $I$ | $X$ | $I$ | $I$ | $I$ |
| $I$ | $I$ | $X$ | $I$ | $I$ | $X$ |
| $I$ | $I$ | $X$ | $X$ | $X$ | $X$ |
| $Z$ | $Z$ | $Z$ | $I$ | $I$ | $Z$ |

1. Identify qubits between two codes.
2. Leave $X$-stabilizers and logical operators unchanged.

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{array}{cccccc} I & X & X & & & \\ X & I & X & & & \\ Z & Z & Z & & & \\ & & X & I & I & X \\ & & X & X & X & X \\ & & Z & I & I & Z \end{array}$$

$$\begin{array}{cccccc} I & X & X & I & I & I \\ X & I & X & I & I & I \\ I & I & X & I & I & X \\ I & I & X & X & X & X \\ Z & Z & Z & I & I & Z \end{array}$$

1. Identify qubits between two codes.
2. Leave $X$-stabilizers and logical operators unchanged.
3. Modify $Z$-stabilizers and logical operators to commute with the $X$ stabilizers.

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$
\begin{array}{cccccc}
I & X & X & & & \\
X & I & X & & & \\
Z & Z & Z & & & \\
 & & X & I & I & X \\
 & & X & X & X & X \\
 & & Z & I & I & Z
\end{array}
$$

$$
\begin{array}{cccccc}
I & X & X & I & I & I \\
X & I & X & I & I & I \\
I & I & X & I & I & X \\
I & I & X & X & X & X \\
Z & Z & Z & I & I & Z
\end{array}
$$

1. Identify qubits between two codes.
2. Leave $X$-stabilizers and logical operators unchanged.
3. Modify $Z$-stabilizers and logical operators to commute with the $X$ stabilizers.
    - The only way to modify the $Z$-stabilizers/logical operators is to weld them.

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{array}{cccccc}
I & X & X & & & \\
X & I & X & & & \\
Z & Z & Z & & & \\
 & & X & I & I & X \\
 & & X & X & X & X \\
 & & Z & I & I & Z
\end{array}$$

$$\begin{array}{cccccc}
I & X & X & I & I & I \\
X & I & X & I & I & I \\
I & I & X & I & I & X \\
I & I & X & X & X & X \\
Z & Z & Z & I & I & Z
\end{array}$$

1. Identify qubits between two codes.
2. Leave $X$-stabilizers and logical operators unchanged.
3. Modify $Z$-stabilizers and logical operators to commute with the $X$ stabilizers.
    - The only way to modify the $Z$-stabilizers/logical operators is to weld them.

- This can easily be generalized to subsystem codes where one simply welds gauge generators, treating them as if they were logical operators on the gauge qubits.

# Welding

Example of welding two stabilizer codes with a Z-weld:

$$\begin{matrix} I & X & X \\ X & I & X \\ Z & Z & Z \end{matrix}$$

$$\begin{matrix} X & I & I & X \\ X & X & X & X \\ Z & I & I & Z \end{matrix}$$

$$\begin{matrix} I & X & X & I & I & I \\ X & I & X & I & I & I \\ I & I & X & I & I & X \\ I & I & X & X & X & X \\ Z & Z & Z & I & I & Z \end{matrix}$$

- We can weld in such a way that the stabilizer generators remain local and/or low weight.

# Welding

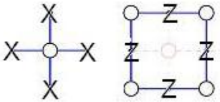
Example of welding two stabilizer codes with a Z-weld:

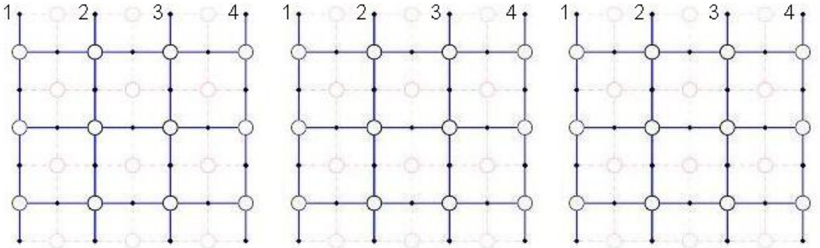
$$
\begin{array}{cccccc}
I & X & X & & & \\
X & I & X & & & \\
Z & Z & Z & & & \\
 & & X & I & I & X \\
 & & X & X & X & X \\
 & & Z & I & I & Z
\end{array}
$$

$$
\begin{array}{cccccc}
I & X & X & I & I & I \\
X & I & X & I & I & I \\
I & I & X & I & I & X \\
I & I & X & X & X & X \\
Z & Z & Z & I & I & Z
\end{array}
$$

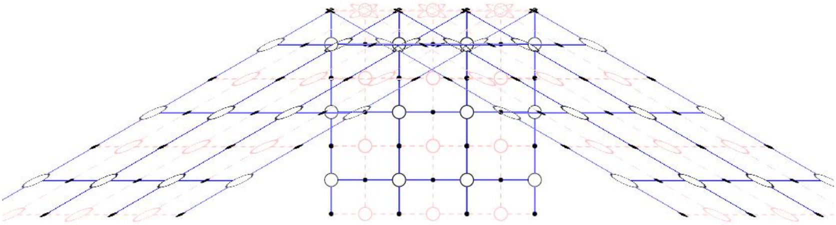
- We can weld in such a way that the stabilizer generators remain local and/or low weight.
- We can design the shape of the logical operators by alternating $X$ and $Z$ welds.

# Welded surface codes

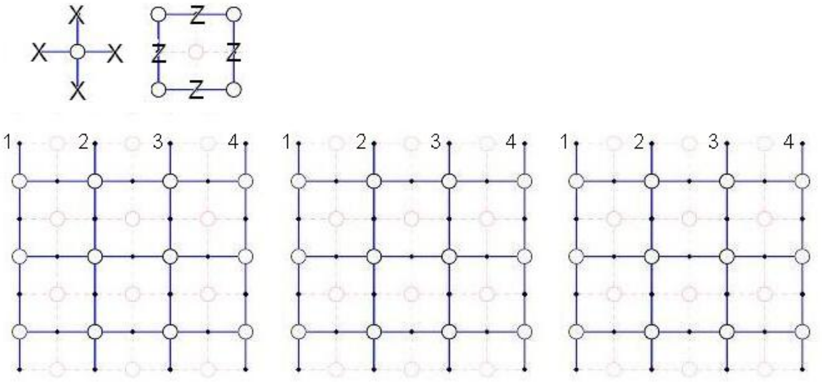


# Welded surface codes

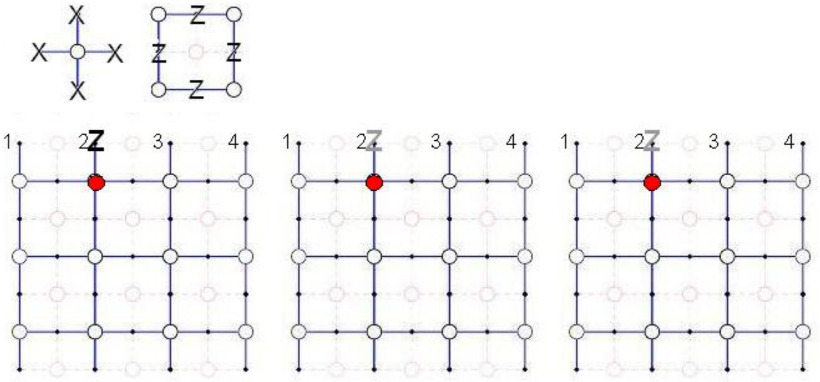


# Welded surface codes



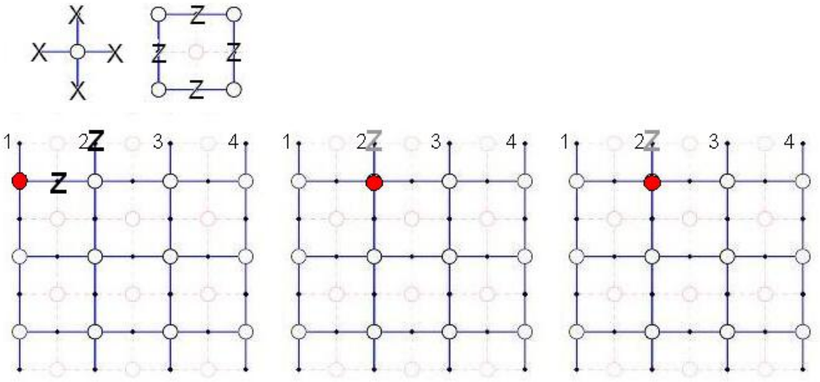

- $Z$-stabilizers get welded together.

# Welded surface codes



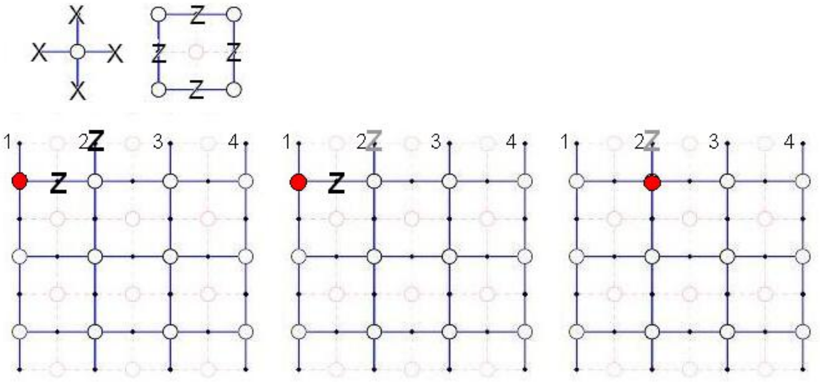

- $Z$-stabilizers get welded together.

# Welded surface codes



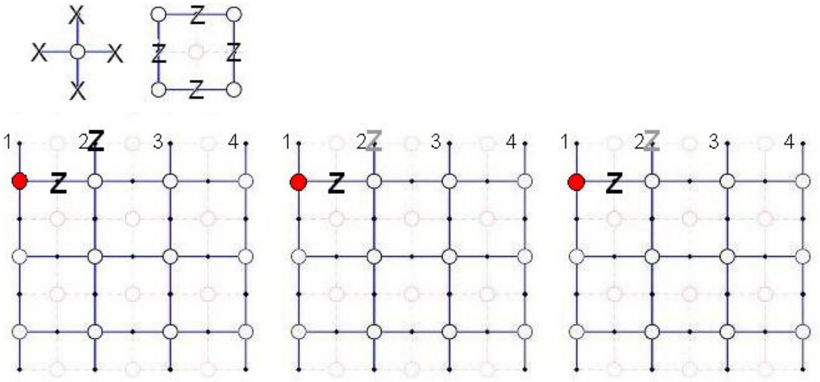

- $Z$-stabilizers get welded together.

# Welded surface codes



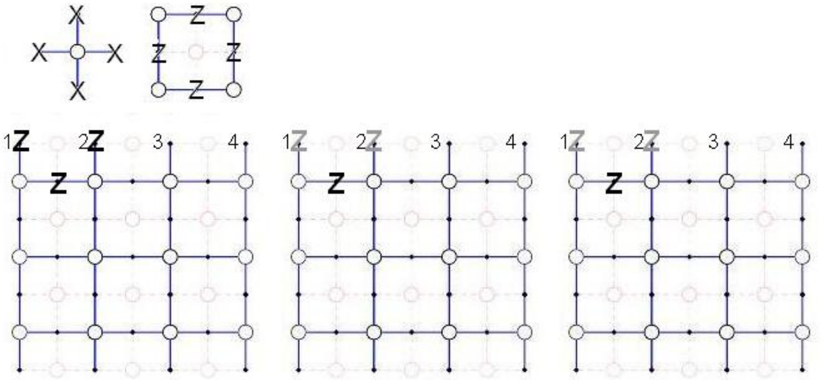

- $Z$-stabilizers get welded together.

# Welded surface codes



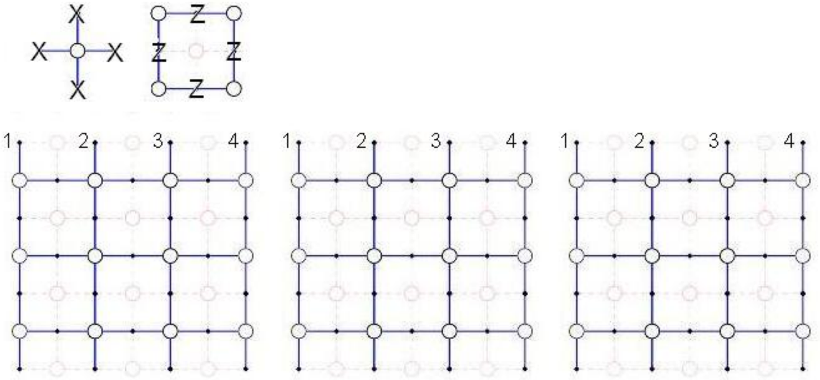

- $Z$-stabilizers get welded together.

# Welded surface codes



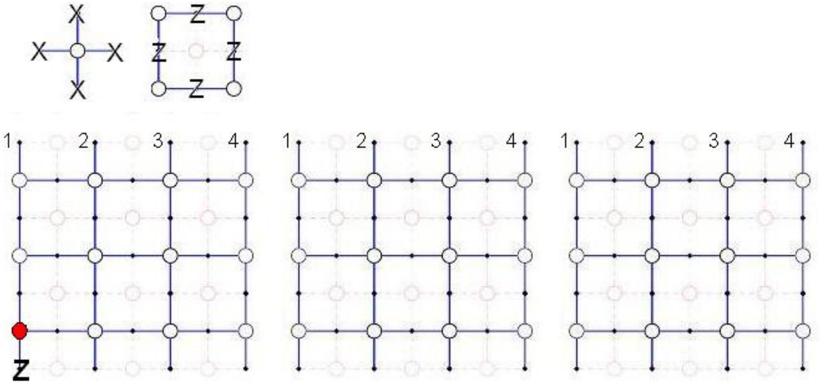

- $Z$-stabilizers get welded together.

# Welded surface codes



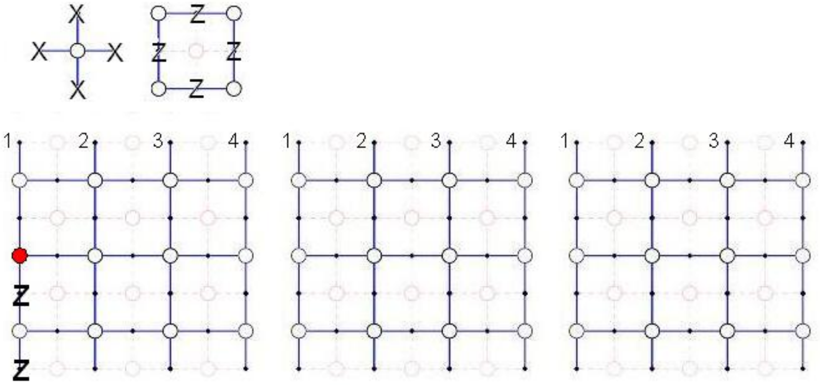

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



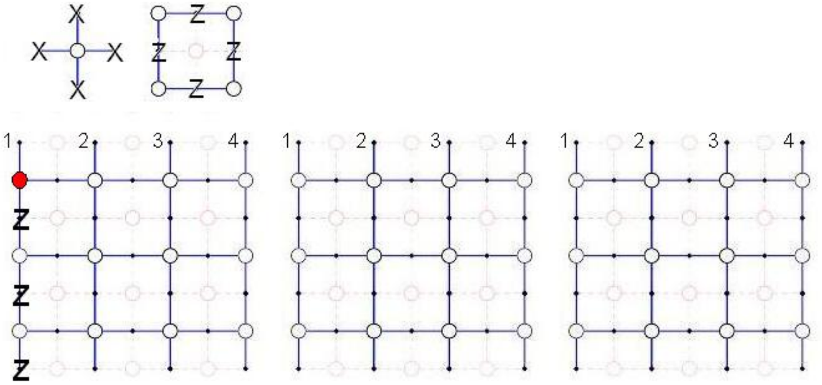

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



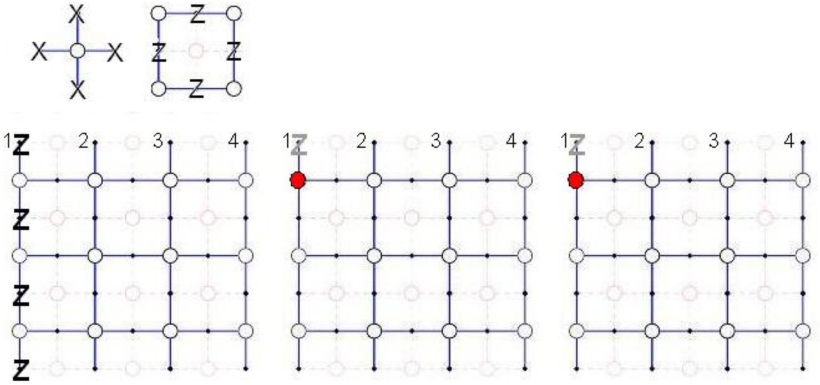

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



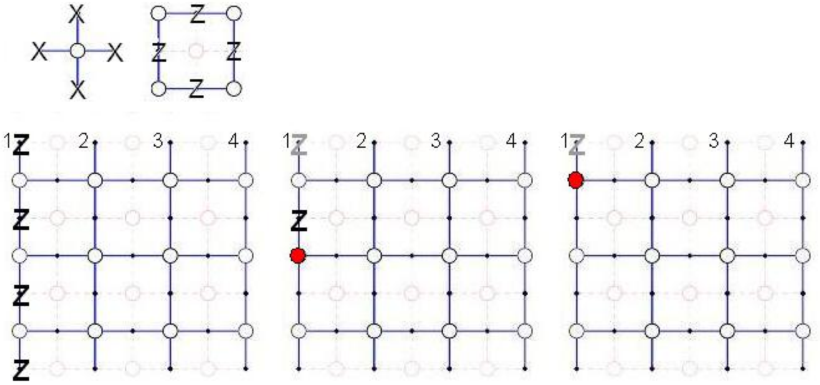

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



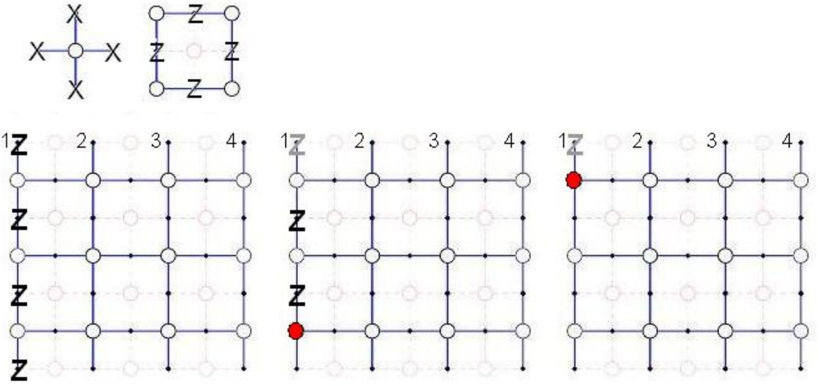

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



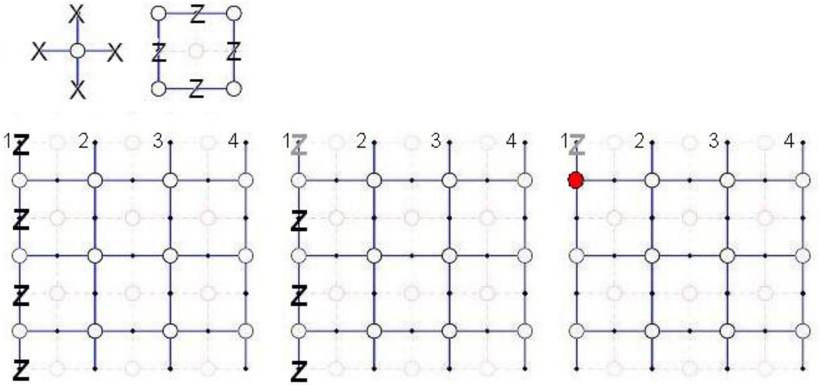

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



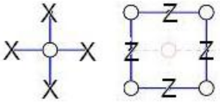

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

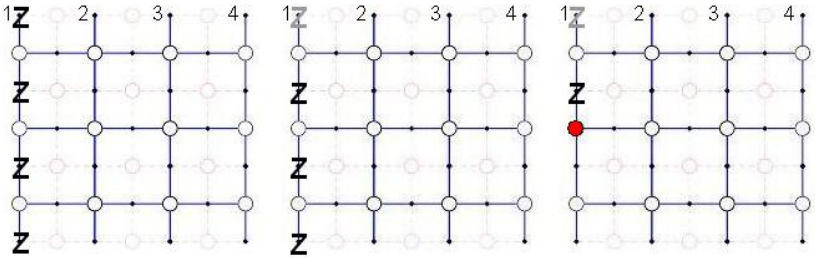
# Welded surface codes




- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

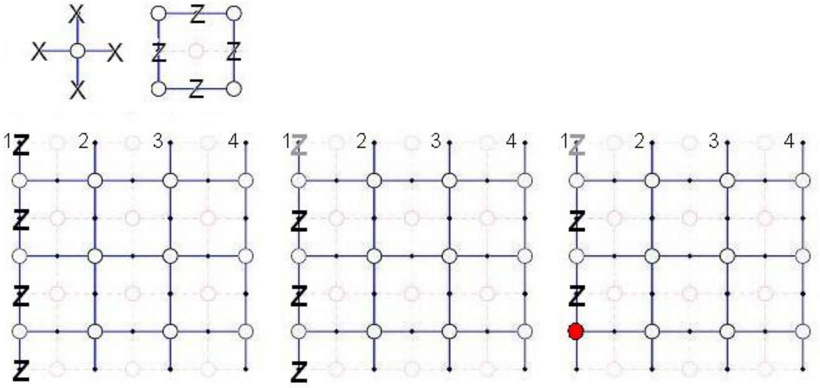
# Welded surface codes



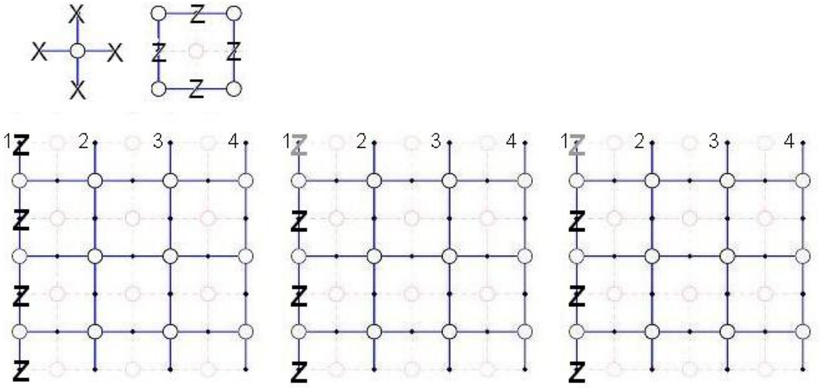

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



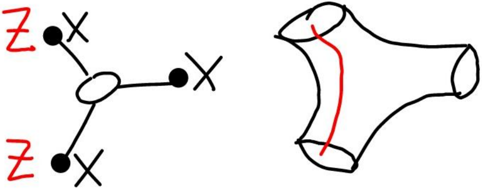

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

# Welded surface codes



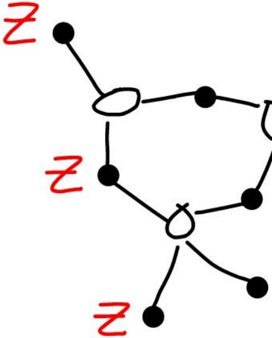

- $Z$-stabilizers get welded together.
- $Z$-strings get welded together.
- The energy barrier of the logical $Z$-operator goes up by one.

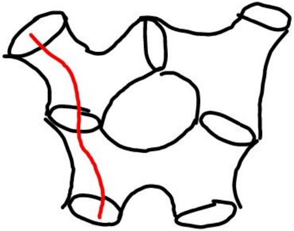
# Generalization of welded solid codes



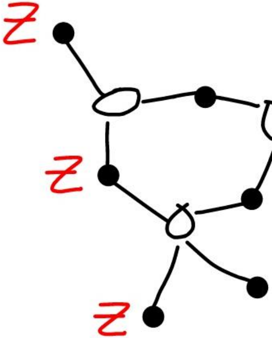

- A solid code with n rough edges behaves like a stabilizer on n qubits.

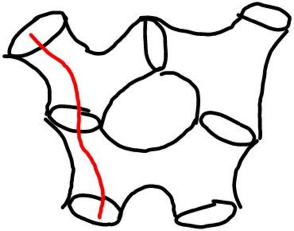
# Classical linear code to topological quantum code



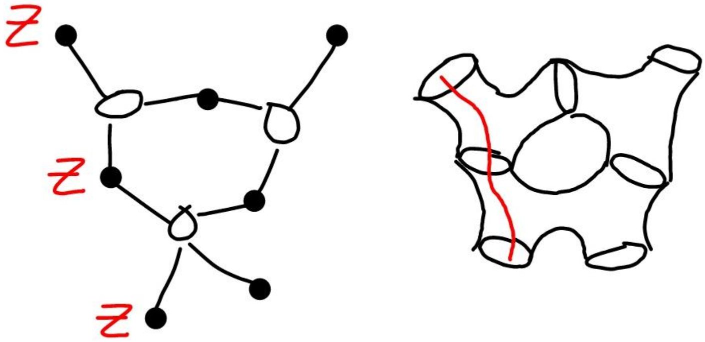
# Classical linear code to topological quantum code



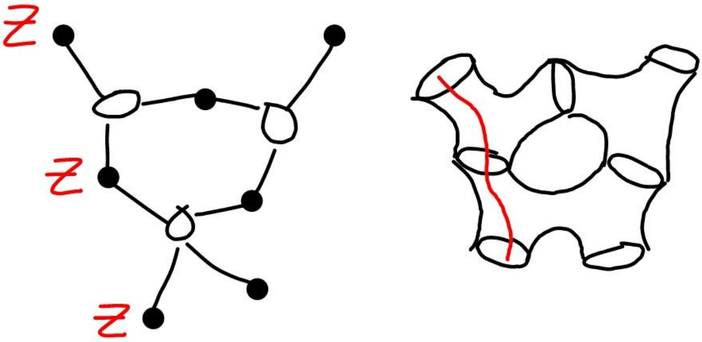
- 3-d local classical $(n, k, d)$ code $\rightarrow$ quantum $(nd^{3/2}, k, d)$ code.

# Classical linear code to topological quantum code




- 3-d local classical $(n, k, d)$ code $\to$ quantum $(nd^{3/2}, k, d)$ code.
- Local low-weight generating set for the above quantum code.

# Classical linear code to topological quantum code




- 3-d local classical $(n, k, d)$ code $\rightarrow$ quantum $(nd^{3/2}, k, d)$ code.
- Local low-weight generating set for the above quantum code.
- The membrane operators are protected by an energy barrier.

# Outlook

# Outlook

- Can we make a self-correcting quantum memory in 3-d?

# Outlook

- Can we make a self-correcting quantum memory in 3-d?
- How good are the welded solid codes in practice?

# Outlook

- Can we make a self-correcting quantum memory in 3-d?
- How good are the welded solid codes in practice?
- What about welding non-surface codes?

# Outlook

- Can we make a self-correcting quantum memory in 3-d?
- How good are the welded solid codes in practice?
- What about welding non-surface codes?
- What about welding non-CSS codes?

# Outlook